四川師範大學
SICHUAN NORMAL UNIVERSITY

第5章　数组和广义表

DATA STRUCTURE

计算机科学学院　廖雪花

本章内容简介

数组和广义表

- 5.1 数组的定义
- 5.2 数组的顺序表示和实现
- 5.3 矩阵的压缩存储
- 5.4 广义表的定义
- 5.5 广义表的存储结构

5.3 矩阵的压缩存储

廖雪花　LiaoXuehua

四川師範大學
SICHUAN NORMAL UNIVERSITY

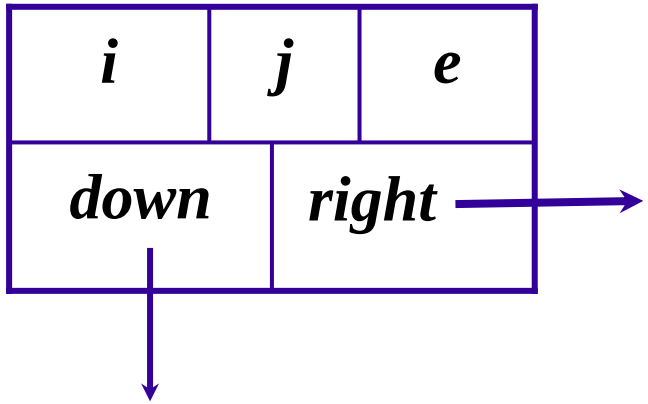
5.3.2 稀疏矩阵

- 几种压缩存储方案
  - 三元组顺序表
  - 带行逻辑链接的顺序表
  - 十字链表

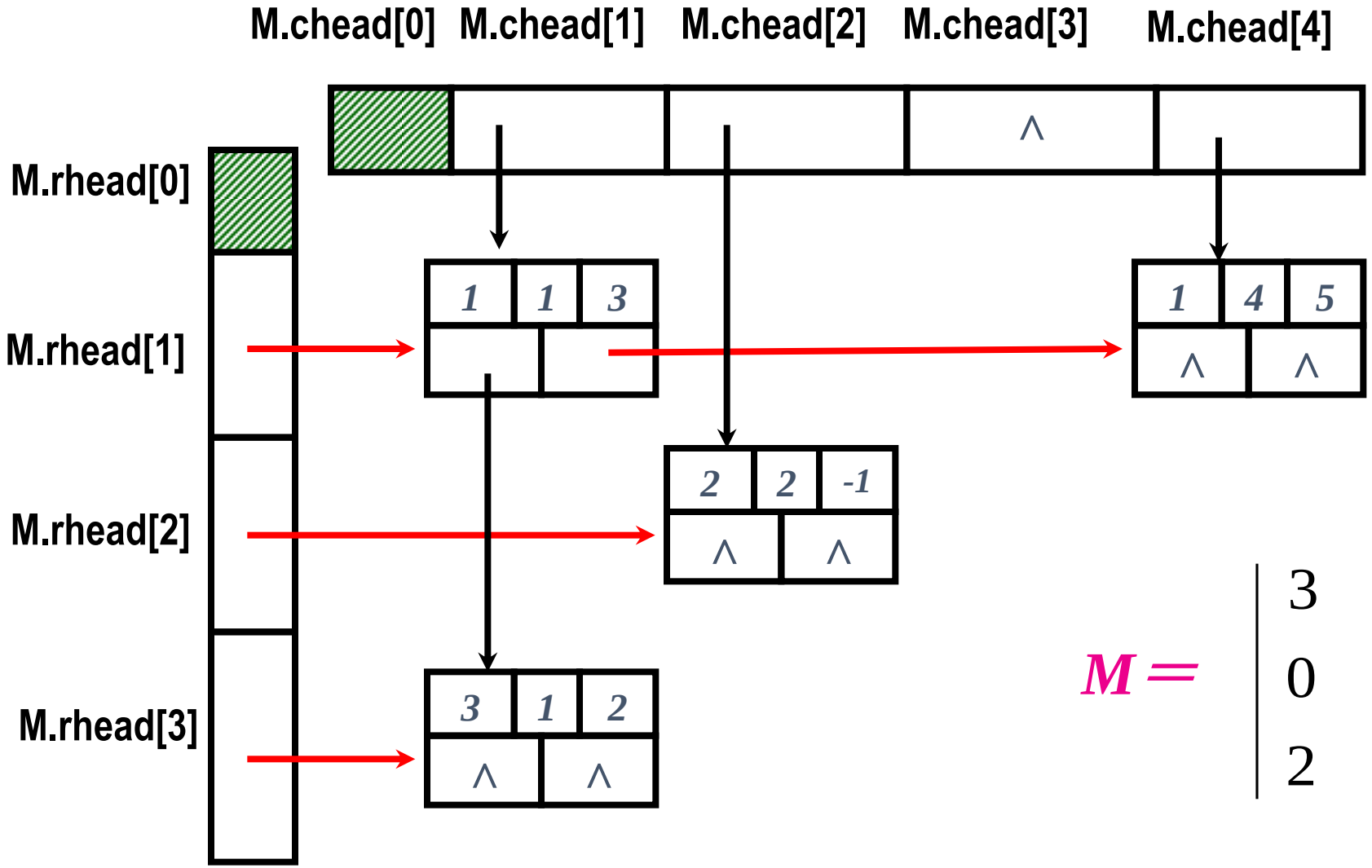
三、十字链表(orthogonal list)

- 结点结构定义

```
typedef struct OLNode{
        int i,j;
        ElemType e;
        struct OLNode *right,*down ;
}OLNode,*OLink;
typedef struct{
        Olink *rhead,*chead;
        int mu,nu,tu;
}CrossList
```



示例：　M.mu=3 M.nu=4 M.tu=4



$$M = \begin{vmatrix} 3 & 0 & 0 & 5 \\ 0 & -1 & 0 & 0 \\ 2 & 0 & 0 & 0 \end{vmatrix}$$

三、十字链表(orthogonal list)

- 建立十字链表表示的稀疏矩阵M

```
Status CreateSMatrix_OL(CrossList &M){
    scanf(&m,&n,&t);
    M.mu=m;M.nu=n;N.tu=t;
    if(!(M.rhead=(OLink*)malloc((m+1)*sizeof(OLink))))
            exit OVERFLOW;      // 申请行头指针向量
    if(!(M.chead=(OLink*)malloc((n+1)*sizeof(OLink))))
      exit OVERFLOW;            //申请列头指针向量
    M.rchead[ ]=M.chead[ ]=NULL; //初始化各行列链表为空表
```

建立十字链表表示的稀疏矩阵M

```
for (k=1;k<=t;k++){
  scanf(&i,&j,&e);
 if (!(p=(Olink)malloc(sizeof(OLNode)))) exit OVERFLOW;
 p->i=i;p->j=j;p->e=e;
  if (M.rhead[i] == NULL || M.rhead[i]->j > j)
     { p->right=M.rhead[i];M.rhead[i]=p;}
 else //否则寻找插入位置q,并在q后插入
     {  for (q=M.rhead[i];q->right &&q->right->j < j;q=q->right);
       p->right=q->right;q->right=p;
      }
```

建立十字链表表示的稀疏矩阵M

```
if (M.chead[j] == NULL || M.chead[j]->I > i)
   {p->down=M.chead[j];M.chead[j]=p; }
else  //否则寻找插入位置q,并在q后插
   { for (q=M.chead[j];q->down&&q->down->I < i;q=q->down);
      p->down=q->down;q->down=p;
   }
  }//for
  return OK;
}
```

三、十字链表(orthogonal list)

- 思考：

对建立好的十字链表表示的稀疏矩阵M，如何输出？

- 按行主序
- 按列主序
- 按矩阵形式

稀疏矩阵的相加运算

$A'=A+B$

$$a_{ij}' = \begin{cases} ①\ b_{ij} & (a_{ij}=0\ b_{ij}\neq 0) \\ ②\ a_{ij} & (a_{ij}\neq 0\ b_{ij}=0) \\ ③\ a_{ij}+b_{ij} & (a_{ij},b_{ij}\neq 0\ a_{ij}+b_{ij}\neq 0) \\ ④\ 0 & (a_{ij},b_{ij}\neq 0\ a_{ij}+b_{ij}=0) \end{cases}$$

考虑稀疏矩阵A，B均采用十字链表表示，当B加入A中，对于A的十字链表来讲：

①插入一个新结点，数据值为b_{ij}

②不变

③修改结点的数据值为$a_{ij}+b_{ij}$

④删除一个结点

稀疏矩阵的相加运算——实现分析

- 1.基本思想：
  - 从矩阵的第一行开始逐行进行比较，对每一行均从第一个非0元开始比较，令pa，pb分别指向矩阵A和B相同行的当前处理结点。
  - ①若pa->j>pb->j或pa==NULL,则：
    - 在A中插入一个值为pb->e的结点，此时需要修改相应的指针。
  - ② 若pa->j<pb->j，则：
    - pa下移，即pa=pa->right
  - ③若pa->j=pb->j
    - pa->e+pb->e ≠0，则pa->e+=pb->e,其它不变；
    - pa->e+pb->e ==0，则在A中删去pa指向的结点，并修改同行前一结点的right指针，以及同一列前一结点的down指针。

稀疏矩阵的相加运算——实现分析

- 2.算法描述：

（1）pa=A.rhead[1]；pb=B.rhead[1]；pre=NULL;

for(j=1;j<=A.nu;j++) hl[j]=A.chead[j];

（2）若pb ≠NULL(B的当前行未处理完），则重复执行本步骤；

①pa==NULL或pa->j>pb->j,生成新结点p

(行表）if (pre==NULL) A.rhead[p->i]=p;//A中第i行为空

else pre->right=p ; //在pre和pa之间插入p

p->right=pa;pre=p;

(列表）//先找到结点p的同一列的前驱结点，并让hl[pa->j]指向该结点

for (q=hl[p->j];q->down&&q->down->i<pa->i;q=q->down);

if (A.chead[p->j]==NULL) {A.chead[p->j]=p;p->down=NULL;}

else {p->down=hl[p->j]->down;hl[p->j]->down=p;}

稀疏矩阵的相加运算——实现分析

- 2.算法描述：

（1）pa=A.rhead[1]；pb=B.rhead[1]；pre=NULL;

for(j=1;j<=A.nu;j++) hl[j]=A.chead[j];

（2）若pb ≠NULL(B的当前行未处理完），则重复执行本步骤；

①pa==NULL或pa->j>pb->j,生成新结点p

(行表）if (pre==NULL) A.rhead[p->i]=p;//A中第i行为空

else pre->right=p ; //在pre和pa之间插入p

p->right=pa;pre=p;

(列表）//先找到结点p的同一列的前驱结点，并让hl[pa->j]指向该结点

for (q=hl[p->j];q->down&&q->down->i<pa->i;q=q->down);

if (A.chead[p->j]==NULL) {A.chead[p->j]=p;p->down=NULL;}

else {p->down=hl[p->j]->down;hl[p->j]->down=p;}

稀疏矩阵的相加运算——实现分析

- 2.算法描述：

②若pa ≠ NULL或pa->j<pb->j，则 pre=pa; pa=pa->right;

③若pa->j==pb->j,则pa->e+=pb->e;

a） pa->e≠0，pa,pb向右移{pre=pa;pa=pa->right;pb=pb->right;}

b）pa->e==0,删除pa,并作指针的相应修改。

(行表）if (pre==NULL) A.rhead[pa->i]=pa->right;

else pre->right=pa->right

p=pa;pa=pa->right;

(列表）先找到结点p的同一列的前驱结点，并让hl[pa->j]指向该结点

if (A.chead[p->j]==p) {A.chead[p->j]=hl[p->j]=p->down;}

else hl[p->j]->down=p->down;

free(p);

稀疏矩阵的相加运算——实现分析

- 2.算法描述：

（3）若本行不是B的最后一行，则令pa,pb分别指向下一行的第一个非0元，重复第（2）步，否则算法结束。

本节要点

- 稀疏矩阵的压缩存储：
  - ✓ 稀疏矩阵的定
  - ✓ 稀疏矩阵的存储：三元组顺序表、带行逻辑链接的顺序表、十字链表

矩阵转置　矩阵乘法　矩阵加法

- 十字链表：
  - ✓ 稀疏矩阵的十字链表表存储表示
  - ✓ 矩阵的加法

四川師範大學
SICHUAN NORMAL UNIVERSITY

感谢聆听